Morphpublishing
Research
open access academic journal

Peer reviewed

Designing Customer 360-Centric Recommender Systems: A Machine Learning Approach to Optimizing B2C Digital Sales and Engagement Outcomes

Nguyen Minh Khoa[a], Tran Duc Long[b], Pham Van Tuan[c]

Abstract: In contemporary B2C environments, customer interactions are increasingly fragmented across web, mobile, physical channels, and third-party ecosystems. Organizations seek to consolidate these touchpoints into a unified Customer 360 view in order to understand behavioral patterns, align content delivery with latent preferences, and manage engagement at scale with consistent semantics across products and segments. Recommender systems sit at the center of this consolidation effort, mediating how customers discover content, offers, and services, and thereby shaping measurable outcomes such as click-through, conversion, retention, and downstream revenue. Traditional recommendation pipelines, often siloed by channel or product line, are not well aligned with a Customer 360 paradigm in which signals, constraints, and objectives must be integrated at the level of individual identities and their temporal trajectories. This paper develops a technical perspective on designing Customer 360-centric recommender systems for B2C digital sales and engagement optimization, focusing on representation learning for heterogeneous data, multi-objective modeling of business and behavioral outcomes, and architectures that close the loop between model outputs and operational feedback signals. The discussion emphasizes modeling formalisms for feature fusion, ranking, calibration, and counterfactual reasoning under production constraints, including latency, robustness, and governance. The aim is to outline a coherent machine learning approach in which Customer 360 data structures are not auxiliary assets but primary modeling substrates through which recommendation quality, consistency, and controllability can be achieved in a measurable and adaptable manner.

1. Introduction

B2C digital sales and service platforms operate in environments where every customer interaction is both a transactional opportunity and a signal about evolving intent, preferences, and constraints [1]. Over the last decade, these interactions have expanded beyond single-channel web sessions into complex patterns spanning native applications, messaging channels, physical locations, contact centers, and third-party intermediaries. Each of these channels produces logs with distinct schemas, latencies, and noise characteristics, often governed by different operational teams and subject to varying regulatory interpretations. In response, many organizations have defined Customer 360 constructs: integrated profiles that consolidate identifiers, events, and attributes into an ostensibly unified representation of each individual. In practice, however, these constructs are frequently implemented as analytics-oriented data marts or dashboards, optimized for reporting, segmentation, and manual campaign design rather than for driving real-time algorithmic decisions [2]. When recommender systems are layered on top of fragmented or reporting-centric infrastructures, they cannot fully exploit the continuity of customer behavior, and their optimization objectives may diverge from those used to evaluate customer value at an organizational level.

Aspect	Traditional B2C Systems	Customer 360-Centric Systems
Data Sources	Channel-specific logs	Unified cross-channel timeline
Identity	Device/account identifiers	Canonical customer identity graph
Objective	Channel KPIs	Multi-objective across engagement, revenue
Consistency	Fragmented	Global across all touchpoints

Table 1. Comparison between Traditional and Customer 360-Centric Recommender Systems.

Layer	Function	Description
Data	Integration	Maps heterogeneous event streams to unified schema
Model	Representation	Learns embeddings and aggregates from timelines
Policy	Optimization	Defines objectives, constraints, and decision logic
Governance	Control	Enforces feature usage and compliance policies

Table 2. Core Layers in Customer 360-Centric Recommender Design.

Component	Examples	Purpose
Static Features	Age, Segment, Loyalty Tier	Capture long-term attributes
Dynamic Features	Recency, Frequency, Category Affinity	Model short-term behaviors
Embeddings	Event Sequence Encodings	Compress interactions into latent space
Aggregates	Rolling Metrics	Enable hybrid learned + handcrafted signals

Table 3. Feature Composition in Customer 360 States.

A Customer 360-centric approach to recommender system design treats unified customer representations not as byproducts of data warehousing but as operational state variables. In this view, each recommendation decision is

Challenge	Manifestation	Mitigation Strategy
Identity Ambiguity	Conflicting device-user mappings	Probabilistic identity resolution
Temporal Drift	Stale embeddings and aggregates	Hybrid streaming + batch refresh
Regulatory Constraints	Restricted feature usage	Feature lineage and masking layers
Evaluation Bias	Confounded by past policies	Counterfactual and randomized designs

Table 4. Challenges and Mitigations in Customer 360 Deployment.

Representation Type	Encoder Function	Key Advantage
Sequential	$g_\theta((v_t, c_t)_{t=1}^T)$	Captures temporal dependencies
Modal Fusion	$f(s_u^{(1)}, \dots, s_u^{(D)})$	Integrates multi-domain signals
Static Attributes	a_u embeddings	Provides stable personalization anchors
Multi-Task Head	$t_k(x_u, z_i)$	Supports diverse prediction goals

Table 5. Representation Learning Components in Customer 360 Models.

Optimization Dimension	Typical Objective	Unified-State Interpretation
Engagement	Click-through Rate	Immediate interaction utility
Revenue	Conversion Probability	Short-term transaction value
Retention	Churn Risk Reduction	Long-term relationship stability
Compliance	Fairness, Policy Adherence	Constraint-based objective terms

Table 6. Multi-Objective Optimization Dimensions in Customer 360 Frameworks.

conditioned on a structured summary of the customer's historical trajectory, contextual environment, and inferred propensities, as encoded in a shared representation that is maintained consistently across channels. The same foundational state is consulted when generating ranked lists in an e-commerce feed, suggesting content in an application, selecting offers in an outbound campaign, or prioritizing service interventions. This alignment has several implications [3]. First, it introduces a requirement for explicit identity resolution, since the quality of any shared state depends on reliable mapping from devices and accounts to canonical customers, with uncertainty modeled rather than ignored. Second, it necessitates representation learning techniques capable of compressing heterogeneous sequences of events into embeddings and aggregates that support low-latency scoring while preserving information relevant to both engagement and sales outcomes. Third, it raises the question of how multiple objectives and constraints, including revenue, satisfaction proxies, exposure regularization, and compliance conditions, can be expressed in a coherent optimization framework that operates over unified states.

Existing recommender deployments in B2C domains illustrate the challenges of not adopting such a unified perspective. Many systems remain segmented by channel or product line, with independent models trained on local signals such as recent clicks, search queries, or cart contents [4]. These models may perform adequately within narrow scopes, but they induce inconsistencies at the customer level: the same person may receive incompatible recommendations across surfaces, experience repetitive exposure to similar items, or encounter abrupt shifts in personalization when crossing device or session boundaries. Moreover, model evaluation is often conducted separately for each surface using local metrics, making it difficult to reason about cumulative effects on customer lifetime value or strategic objectives. When Customer 360 artifacts exist only in parallel for reporting, any attempt to reconcile system behavior with holistic metrics becomes indirect, relying on offline joins and retrospective analyses. This gap complicates both optimization and governance, because corrective measures must be implemented through fragmented levers rather than through principled changes to shared representations and policies.

A technically grounded Customer 360-centric design addresses these issues by integrating data architecture, model construction, and decision logic around the shared concept of a customer state [5]. At the data level, this entails constructing schemas in which events from all relevant channels are mapped to a canonical timeline per customer, with associated contextual descriptors and catalog references. At the modeling level, it involves training representation learners that consume these timelines to produce state vectors that are stable enough for reuse yet responsive enough to capture shifts in behavior. At the policy level, it requires specifying optimization problems in which the decision variables are recommendation actions and the constraints and objectives are functions of these shared states, making explicit how exposures influence immediate and downstream metrics. Such a design does not assume that a single monolithic model controls all recommendations; rather, it allows multiple models and policies to read from and write to the same underlying state, coordinating through shared abstractions instead of bespoke interfaces. [6]

An important aspect of this perspective is its neutrality with respect to domain-specific assumptions. The mechanisms for constructing Customer 360 profiles, encoding trajectories, and optimizing recommendations can be described without reference to particular industries, content types, or interface patterns. This neutrality is advantageous because it supports reuse of core infrastructure and methodological components across varied contexts, including retail, media, travel, finance, and telecommunications, where specific constraints and objectives differ but the underlying need for coherent, identity-resolved personalization is similar. By formulating models and architectures in terms of general-purpose constructs such as event sequences, item attributes, and multi-objective utilities, organizations can adapt the same conceptual framework while instantiating different policy parameters, feature subsets, or governance rules. This separation of concerns encourages clarity: technical implementations are evaluated based on how well they satisfy formally stated requirements rather than on implicit narratives about user experience. [7]

The introduction of Customer 360-centric recommenders also foregrounds the interplay between short-term interaction optimization and long-term relationship management. Traditional recommenders frequently target proximal metrics such as click-through rate or immediate conversion probability, which are easy to measure and convenient for gradient-based optimization. However, these metrics may not align with durable engagement, recurring purchases, or risk considerations. A unified customer state, informed by historical trajectories and cross-channel behavior, makes it feasible to incorporate proxies for longer-term outcomes into the modeling process, such as indicators of churn risk, value volatility, or responsiveness to particular communication patterns. While such proxies remain approximations, their inclusion within a structured state representation opens a path toward policies that are more sensitive to the temporal dynamics of customer relationships, without requiring fully specified lifetime value models or complex reinforcement learning pipelines in every case. [8]

Temporal and causal structures are not incidental details in this context; they are embedded in the definition and use of Customer 360 states. Every recommendation made at a given time depends on past events and in turn influences future states by shaping exposures and responses. Logged data thus reflect the composition of historical policies and customer behavior, creating dependencies that affect how new models can be trained and evaluated. A Customer 360-centric paradigm makes these dependencies more visible by assigning recommendations and outcomes to persistent states rather than to isolated sessions or devices. This explicitness does not eliminate confounding or feedback loops, but it provides the scaffolding needed for counterfactual reasoning, propensity-based estimators, and careful experimental design [9]. Without such scaffolding, attempts to adjust recommendation logic in response to observed patterns risk conflating structural effects with artifacts of logging or identity stitching.

Another rationale for emphasizing Customer 360-centric design is its impact on explainability, governance, and risk management. When recommendation behavior is driven by a fragmented collection of channel-specific heuristics and models, tracing the origin of a particular exposure pattern can be difficult, especially when identity resolution is handled differently across systems. Centralizing state definitions and decision interfaces simplifies such tracing by reducing the number of distinct mechanisms that must be examined [10]. It also enables enforcement of feature-level policies and constraints at a single point of control, instead of relying on synchronous updates across multiple systems. This structural consolidation is relevant for compliance with data protection regulations and internal standards, but it is also relevant for technical robustness: centralized constraints reduce the likelihood that unintended dependencies on sensitive attributes or unstable proxies will arise in uncoordinated components.

A further dimension of the introduction concerns the relationship between Customer 360-centric recommenders and broader organizational MLOps practices. In many B2C settings, recommendation models coexist with systems for fraud detection, credit or risk scoring, inventory forecasting, and marketing attribution, all of which consume overlapping data and potentially influence how customers experience the platform. Treating Customer 360 states as a shared primitive encourages unification of feature pipelines, monitoring strategies, and deployment controls across these use cases [11]. This does not imply that all models must share the same architecture, but it does imply that they can draw from a consistent representation of customer behavior and attributes, reducing redundancy and lowering the risk of subtle inconsistencies. Moreover, unified states facilitate multi-system analyses, such as understanding how changes in recommendation policies interact with incentives or pricing strategies, because the same identifiers and temporal references underpin observational data for all components.

Finally, it is necessary to recognize that the adoption of Customer 360-centric recommender architectures is shaped by practical constraints including legacy systems, data quality limitations, staffing, and incremental risk tolerance. The concepts developed in this paper are intended to be composable under such constraints. Organizations may initially rely on relatively coarse identity stitching and simple sequence summaries, yet still benefit from structuring their systems around explicit state variables and multi-objective policies [12]. Over time, more refined identity resolution, richer embeddings, and advanced evaluation methods can be introduced without invalidating earlier design choices, provided that those choices were articulated in terms of general abstractions. This incremental compatibility is important because it reduces the need for disruptive transitions and allows operational evidence to inform each successive refinement. The remainder of the paper therefore treats Customer 360-centric recommender design as an evolving engineering discipline, in which formal models, data infrastructures, and governance frameworks co-develop, anchored by the consistent use of unified customer states as the central objects of reasoning.

An additional motivation for a detailed introduction is to clarify how Customer 360-centric recommender systems relate to long-standing themes in personalization research, such as collaborative filtering, content-based modeling, and hybrid architectures. These classical approaches typically operate over matrices or graphs that encode interactions between users and items, sometimes augmented with contextual features [13]. The Customer 360 perspective extends this by insisting that the definition of a user itself be grounded in an explicit, identity-resolved and temporally structured construct, rather than in an abstract index whose meaning may drift as identifiers change.

It also emphasizes that side information about users is not a static set of attributes but a stream of events arriving from multiple operational systems, each with its own reliability and policy constraints. Integrating such streams into a disciplined state representation alters how models are regularized, how cold-start conditions are treated, and how shifting catalogs and acquisition channels are incorporated. By situating traditional techniques within this broader framing, the introduction aims to prevent a narrow focus on algorithmic novelty from obscuring the importance of data semantics and system integration, which often determine the feasibility and stability of large-scale deployments more strongly than marginal gains in predictive accuracy on isolated benchmarks.

Equally significant is the tension between the richness of Customer 360 representations and the operational constraints of real-time recommendation. Detailed identity graphs, multi-channel histories, and high-dimensional embeddings increase modeling capacity but also impose latency, memory, and throughput demands on serving infrastructure. A Customer 360-centric introduction must therefore acknowledge that not all theoretically desirable signals can be used symmetrically at inference time. Practical designs distinguish between core state components that are guaranteed to be fresh and efficiently retrievable, and auxiliary components that may be updated less frequently or consulted only in specific workflows. This distinction shapes both algorithm choice and interface design: ranking models are built to depend primarily on the stable subset of features, while side-information is incorporated through calibration layers, post-processing adjustments, or offline analyses [14]. Highlighting this trade-off at the outset clarifies that the subsequent sections consider feasibility under realistic deployment conditions, not merely under idealized assumptions where all customer information is instantaneously available and perfectly reliable.

Rather than advocating a single metric, algorithm, or vendor solution, the focus is on identifying structural properties that any implementation must resolve: how identities are defined and maintained, how histories are encoded, how competing objectives are formalized, how temporal feedback is handled, and how constraints are enforced. By foregrounding these properties, subsequent sections aim to provide a technical vocabulary and set of modeling patterns that can be instantiated with different tools and infrastructures while preserving conceptual coherence. The intention is that, once these foundations are in place, discussions about specific models or optimizations can proceed with clearer reference points and fewer ambiguities about the meaning of personalization, performance, and control in Customer 360-centric recommender systems.

2. Customer 360 Data Foundations for B2C Recommender Systems

A Customer 360-centric recommender system assumes the existence of an integrated data model that organizes all observable interactions and attributes around stable customer identifiers [15]. Let u index customers and t index discrete event times across channels. For each (u, t), the platform may observe events $e_{u,t}$ belonging to domains such as browsing, transactions, campaign interactions, search, service contacts, and device or location signals. Each event can be mapped to a feature vector in a domain-specific space, yet naive concatenation across domains is rarely suitable due to sparsity, temporal imbalance, and spurious correlations.

Identity resolution is the basis of Customer 360 construction. Let there be raw identifiers i (cookies, device IDs, logins, loyalty IDs) and a mapping function that assigns them to canonical customers. A simplified abstraction introduces a probabilistic mapping where each identifier i is linked to customer u with confidence $p(u \mid i)$ [16]. Deterministic stitching corresponds to assigning each i to the most probable u, whereas more conservative regimes weight events in training by association confidence. While the operational identity graph can be complex, representing its uncertainty in model inputs can reduce brittleness to resolution errors without introducing additional structural assumptions.

Customer 360 states need to encode both static and dynamic features. Static attributes, such as age brackets or long-term segment memberships, are subject to governance constraints and may be partially unavailable or noisy [17]. Dynamic attributes, such as frequency of visits, recency of purchases, and evolving category affinities, reflect short-horizon behavior. A common strategy is to maintain rolling aggregates for each customer, such as counts, sums, or time-decayed statistics. However, when designing a machine learning-centric architecture, these aggregates are subsumed by learned embeddings that compress sequences of events. The Customer 360 store then exposes embeddings and structured aggregates side by side, enabling models to mix parametric summaries with directly learned representations.

From a systems perspective, the data foundation must support streaming and batch paths [18]. Real-time updates to the Customer 360 state enable near-instant adaptation of recommendations after key events such as purchases or churn signals, while batch recomputations allow for re-embedding long histories and re-estimating slow-moving features. Consistency between these paths is essential: models should be trained on data representations that closely match those used at inference time. Misalignment between offline and online encodings can cause systematic drift in recommendation quality and complicate the interpretation of lift experiments.

Governance constraints shape which signals are admissible and how they can be combined. Certain attributes may be restricted to specific use cases, while others require de-identification or aggregation [19]. Customer 360-centric recommenders therefore need explicit mechanisms to mask or transform sensitive features, and to document feature lineages for auditability. These constraints can be integrated into feature generation layers such that downstream modelers operate on sanctioned feature views without ad hoc handling. This integration reduces the risk of leakage from restricted attributes into recommendation policies and promotes stable behavior under changing regulatory guidance.

Finally, Customer 360 architectures introduce an implicit causal structure: recommendations depend on aggregated past exposures and responses, which themselves depend on earlier policies. Any analysis of effect sizes, uplift, or lifetime value must therefore acknowledge policy-induced confounding [20]. Even before formal causal modeling is introduced, a carefully instrumented exposure logging scheme with consistent identifiers and timestamps is required to enable unbiased evaluation strategies using randomized experiments or quasi-experimental methods. Customer 360 data models that preserve such structure offer a practical foundation for the modeling techniques discussed in subsequent sections.

3. Representation Learning and Feature Fusion in Customer 360-Centric Recommenders

Given a Customer 360 data backbone, the next step is to construct representations that can capture complex preference patterns while remaining computationally tractable. Let x_u denote the Customer 360 feature vector or embedding exposed to recommendation models for customer u, and let $y_{u,i}$ represent the target signal associated with recommending item i to u, such as click or conversion. The objective is to derive x_u from raw events in a way that is expressive yet stable across time, channels, and tasks.

Consider sequences of interactions for each customer, such as viewed items $(v_1, \dots, v_T)$ and associated context $(c_1, \dots, c_T)$. A sequence encoder maps these to an embedding s_u. A basic formulation employs a recurrent or self-attentive encoder [21]. In abstract form, define

$$s_u = g_\theta\big((v_t, c_t)_{t=1}^{T}\big), \tag{1}$$

where g_θ denotes a parameterized function that processes the ordered events. To respect the 8cm width constraint, symbol sets and arguments are kept concise, and long textual descriptions of inputs are handled outside the formal expression. The embedding s_u can be concatenated or combined with static attributes a_u and cross-channel aggregates r_u to yield

$$x_u = h_\phi(s_u, a_u, r_u), \tag{2}$$

where h_ϕ is chosen to be a lightweight network ensuring efficient inference latency.

Feature fusion over multiple domains is handled through modality-specific encoders. Let $s_u^{(d)}$ be the embedding derived from domain d, such as web browsing or transactional history. A fusion function defines

$$x_u = f\big(s_u^{(1)}, s_u^{(2)}, \ldots, s_u^{(D)}\big), \tag{3}$$

with f implemented as a gated or attention-based combiner that can attenuate unreliable modalities [22]. Regularization strategies aim for smoothness across customers with similar histories and for invariance to shifts in observation density. For instance, embeddings can be constrained so that their norms remain within a calibrated band, avoiding exploding magnitudes for heavy users relative to light users.

Item representations, denoted z_i, are derived from content metadata, catalog structure, and interaction patterns. A typical scoring function for relevance is

$$s_{u,i} = \sigma\big(x_u^\top z_i\big), \tag{4}$$

where σ is a monotonic link, such as a logistic or identity mapping. This low-rank formulation allows efficient candidate scoring while leveraging Customer 360 features within x_u. More expressive architectures introduce non-linear interactions: [23]

$$s_{u,i} = q_\psi(x_u, z_i), \tag{5}$$

with q_ψ modeled as a small feed-forward network. Capacity control and calibration are used to prevent unstable scores induced by minor perturbations in upstream features.

The multi-task nature of B2C optimization suggests sharing the Customer 360 representation across objectives such as click propensity, conversion likelihood, churn risk, or content completion. Let targets $y_{u,i}^{(k)}$ index multiple outcomes. A shared encoder produces x_u, and task-specific heads output predictions

$$\hat{y}_{u,i}^{(k)} = t_k\big(x_u, z_i\big). \tag{6}$$

Joint training with appropriate loss weighting encourages the representation to encode features informative across tasks without manually engineering task-specific aggregations. Careful construction of the loss, discussed later, avoids dominance by dense but low-signal tasks or by rare high-value events.

Representation learning is tightly coupled with data freshness and drift management [24]. Embeddings must be periodically updated to reflect new behaviors, yet excessive recomputation can be operationally expensive and may introduce instability. A practical approach maintains streaming updates for short-term features while refreshing longer-term components on a slower cadence. Models are trained to tolerate slight staleness in x_u by incorporating realistic delays into the training pipeline, thereby aligning optimization with actual serving conditions.

4. Multi-Objective Recommendation Modeling for Sales and Engagement

Customer 360-centric recommenders rarely optimize a single metric. Platforms typically monitor engagement rates, revenue, catalog coverage, session depth, and various risk or compliance indicators [25]. Formulating recommendation as a multi-objective optimization problem clarifies trade-offs between these criteria within a consistent mathematical structure.

Consider a ranked list π_u of items shown to user u. Let $r_{u,i}$ denote an immediate reward component, such as probability of click or purchase, and let $c_{u,i}$ capture auxiliary costs or penalties, such as excessive repetition or promotion of low-margin items. A basic stochastic objective is

$$J = \mathbb{E}\big[R(\pi_u)\big], \tag{7}$$

where $R(\pi_u)$ aggregates item-level utilities at serving time. For multi-objective settings, define component utilities $R^{(k)}(\pi_u)$ and consider

$$J_\lambda = \sum_k \lambda_k \, \mathbb{E}\big[R^{(k)}(\pi_u)\big], \tag{8}$$

where $\lambda_k \geq 0$ encode relative preferences among objectives. In practice, $R^{(1)}$ may represent predicted incremental revenue, $R^{(2)}$ engagement or satisfaction proxies, and $R^{(3)}$ catalog or policy constraints expressed as soft penalties. The choice of λ_k is guided by business configuration rather than model behavior alone.

For click or conversion modeling, let binary outcome $y_{u,i} \in \{0, 1\}$ measure whether user u engaged with item i after exposure. A parametric model with parameters θ outputs score $s_{u,i}$ and estimated probability

$$p_\theta(y_{u,i} = 1 \mid x_u, z_i) = \sigma\big(s_{u,i}\big). \tag{9}$$

A cross-entropy loss captures the primary objective. To incorporate additional objectives, the training loss can be extended as

$$L(\theta) = [26] \sum_{(u,i)} \ell\big(y_{u,i}, s_{u,i}\big) + \sum_m \alpha_m \, \Omega_m(\theta), \tag{10}$$

where $\Omega_m(\theta)$ implement regularization terms that approximate constraints such as exposure diversity or margin thresholds. For example, one may penalize variance of cumulative exposure across item groups or enforce soft bounds on average predicted margin.

In dynamic settings, myopic optimization of immediate probabilities may diverge from long-run revenue or retention. A sequential formulation treats each recommendation as an action in a Markov decision process with state x_u and reward signal capturing both short-term and proxy long-term outcomes. Policy gradient or actor-critic methods can then be employed, while still using supervised models as baselines. Let π_θ denote the stochastic policy over ranked items. An approximate objective is

$$J(\theta) = [27] \mathbb{E}_{\pi_\theta}\big[G\big], \tag{11}$$

where G is a discounted sum of rewards estimated from logged data or controlled experiments. Because full reinforcement learning pipelines can be sensitive to bias and variance under off-policy evaluation, many production architectures adopt hybrid strategies, using supervised propensity models constrained by policy-level heuristics or simulation-derived penalties to approximate multi-step considerations without fully relying on high-variance estimators.

Multi-objective modeling must remain interpretable to operators who adjust parameters and review outcomes. Clear decomposition of objectives and constraints aids configuration changes when commercial conditions or regulatory requirements shift. For example, if inventory considerations demand a minimum fraction of exposure for a class of items, the model can incorporate a Lagrangian term that penalizes deviations from this constraint, enabling explicit tuning rather than opaque re-training cycles [28]. Customer 360 signals enter these objectives through x_u, enabling nuanced policies such as relaxing frequency caps for high-satisfaction segments or tightening risk-sensitive exposures, without hard-coding per-segment rules.

5. Temporal Dynamics, Causality, and Counterfactual Evaluation

Customer 360-centric recommendation policies necessarily operate in a temporal and causal setting where present actions influence future states and observed data reflect prior policies. Ignoring this structure can result in biased effect estimates, overfitting to historical artifacts, and unstable changes when models are updated. A careful formulation separates three elements: temporal modeling of customer states, causal reasoning about policies, and counterfactual evaluation of candidate models.

Temporal dynamics can be modeled through latent states $h_{u,t}$ summarizing customer u at time t. A generic state update takes the form [29]

$$h_{u,t+1} = F\big(h_{u,t}, e_{u,t}\big), \tag{12}$$

where F may be a recurrent network or a simpler function over event encodings. Recommendation scores at time t for items i depend on $h_{u,t}$ via

$$s_{u,i,t} = q\big(h_{u,t}, z_i\big). \tag{13}$$

The Customer 360 store can retain $h_{u,t}$ or a compressed derivative thereof as part of x_u, making temporal evolution explicit in training and serving. Such stateful models capture seasonality, habit formation, or churn precursors without manually defining multi-window aggregates.

Causal considerations arise because both training and evaluation data reflect the exposure mechanisms of historical recommenders. Let $a_{u,t}$ denote the recommended action (e.g., ordered list), $y_{u,t}$ the resulting outcome, and $\mu(a_{u,t} \mid x_{u,t})$ the logging policy. For an alternative policy π, counterfactual evaluation seeks

$$V(\pi) = [30]\mathbb{E}\big[Y^{\pi}\big], \tag{14}$$

where Y^{π} is the outcome that would be observed if π had been deployed. Logged bandit feedback can be used with inverse propensity weighting or related estimators, provided that propensities under μ are available and positivity conditions approximately hold. A basic inverse propensity estimator is

$$\hat{V}(\pi) = \frac{1}{N}\sum_{n=1}^{N} \frac{\mathbb{I}\{a_n = \pi(x_n)\}\, y_n}{\mu(a_n \mid x_n)}. \tag{15}$$

To constrain expression width, the notation omits secondary indices and relies on compact symbols. Variance reduction techniques, including self-normalization and doubly robust estimators, are applicable but require consistent propensity and outcome models.

Customer 360 attributes may themselves be affected by prior actions, which complicates causal interpretation [31]. For instance, loyalty status and engagement embeddings depend on historical recommendations and promotions. Using such variables as confounders without accounting for their policy dependence can bias estimates of treatment effects. One pragmatic approach is to distinguish between pre-treatment and post-treatment features in modeling uplift for specific interventions, keeping only features that are not consequences of the intervention in the adjustment set. Another is to model state transitions explicitly, as in structural models where

$$h_{u,t+1} = F\big(h_{u,t}, a_{u,t}, y_{u,t}\big), \tag{16}$$

and to analyze policies in terms of induced trajectories. [32]

Temporal credit assignment is relevant when objectives include long-term metrics such as repeat purchase or subscription retention. Direct optimization on long delays is often impractical, so surrogate targets are used.

Customer 360 representations can include estimates of expected future value, denoted $V_{u,t}$, derived from survival models or recurrent predictors. A simple parametric survival model for churn hazard $\lambda(t \mid x_{u,t})$ yields survival function

$$S(t \mid x_{u,t}) = \exp\left(-\int_0^t \lambda(\tau \mid x_{u,\tau})\, d\tau\right), [33] \tag{17}$$

which can be approximated discretely. These quantities can serve as auxiliary labels guiding recommenders toward actions associated with lower churn risk, while leaving room for experimentation to validate assumptions.

Evaluation protocols in Customer 360-centric systems combine randomized experiments and logged-data estimators. Experiments remain a primary instrument for unbiased measurement of incremental effects, but their design depends on stable identity resolution and consistent state tracking. Assigning experiments at the customer level rather than the device level ensures coherence with Customer 360 semantics [34]. Logged policy data with propensities enable offline screening of candidate models to select those most promising for online trials, thereby reducing experimental costs and limiting exposure to poorly performing variants. Counterfactual evaluation thus forms an integral layer in the model lifecycle, grounded in the structures encoded by the Customer 360 data model.

6. Implementation Patterns and Experimentation Methodology in Customer 360-Centric Recommenders

Designing and operating Customer 360-centric recommender systems requires not only formal models and architectural blueprints but also implementation patterns that allow organizations to integrate these components into existing technology stacks and decision processes. This section discusses such patterns from a neutral and system-oriented perspective, focusing on how abstract constructs such as unified identities, shared embeddings, multi-objective policies, and counterfactual estimators can be instantiated incrementally without assuming a complete greenfield environment. The emphasis is on coordination between data engineering, machine learning, and product operations under constraints of reliability, explainability, and change management [35]. Rather than presenting prescriptive recipes for specific industries, the discussion aims to isolate reusable method fragments that preserve the semantics of Customer 360 constructs and create predictable trajectories for system evolution.

A natural pattern for incremental adoption begins with defining canonical feature views that are derived from raw Customer 360 data but are sufficiently stable to support multiple modeling efforts. These views include base aggregates, coarse-grained sequence summaries, and catalog attributes that have been validated for consistency and compliance. Recommendation teams consume these views as their initial feature space and implement relatively simple models, such as factorization machines or shallow neural networks, to estimate relevance scores. Even at this stage, explicit logging of exposure decisions and propensities is introduced, so that future upgrades to multi-objective or causal methods can leverage historically compatible data [36]. The underlying principle is that establishing disciplined data contracts and logging semantics yields long-term benefits even when models are modest, because it reduces friction when more sophisticated Customer 360-centric architectures are later deployed.

A second pattern introduces shared embedding services as first-class internal products. Instead of embedding logic being tightly coupled to individual recommendation applications, organizations define services that map Customer 360 trajectories into embeddings x_u and catalog entities into embeddings z_i using versioned models. Downstream recommenders, search systems, and marketing tools query these embeddings through stable interfaces, while the embedding service itself evolves according to its own experimentation roadmap. This separation localizes complexity: teams using embeddings can reason in terms of vector spaces and compatibility guarantees, while the embedding team is responsible for training, evaluation, and monitoring [37]. The contract specifies that for a given model version, vectors remain stable over a defined horizon, and that migration to new versions occurs through controlled rollouts with dual-writing capabilities so that downstream consumers can compare behaviors using shadow traffic or offline replays without abrupt changes.

Candidate generation strategies reflect another key implementation decision. Many large-scale systems adopt multi-stage pipelines where inexpensive heuristics or approximate similarity lookups produce a candidate set that is then ranked by more expressive models. In Customer 360-centric designs, candidate generation is modified so that it directly incorporates x_u rather than relying solely on short-term context. For example, approximate nearest neighbor indexes can be constructed in the space of z_i, and candidate retrieval can prioritize items whose z_i align with x_u under a simple score such as

$$s_{u,i} = x_u^\top z_i.$$

This alignment ensures that candidate sets are structurally compatible with ranking models that use the same representation families, reducing the mismatch often observed when retrieval and ranking are optimized independently. Over time, candidate generators themselves can evolve into learned retrieval models trained on Customer 360-aware signals, while retaining fallbacks based on popularity or recency to ensure coverage. [38]

Experimentation methodology is central to validating Customer 360-centric recommenders and adjusting policies with minimal ambiguity. A robust pattern assigns experimental units at the level of canonical customers, aligning with the identity definitions used in the Customer 360 store. Treatment assignment is recorded as part of the event schema, and propensities are captured whenever allocation is not uniform. This structure supports both standard randomized controlled trials and contextual bandit experiments. For example, when evaluating an updated ranking model, the system can allocate a fraction of customers to the new policy while the remainder serve as a control, ensuring that cross-device behavior and temporal consistency are preserved within assignments [39]. Because Customer 360 trajectories persist across experiments, long-horizon effects, such as changes in repeat purchase or subscription continuation, can be assessed without reconstructing identities retrospectively.

Incremental rollout strategies follow a consistent template. New models or policy configurations are first tested in shadow mode, where they receive live traffic and compute recommendations without influencing user experiences. Their outputs, conditioned on Customer 360 features, are logged and compared to the incumbent system using offline metrics and counterfactual estimators that rely on recorded propensities from current policies. This step identifies gross inconsistencies, degenerate behaviors on specific segments, or violations of constraints [40]. Subsequent online experiments expose a small proportion of customers to the new policy, with monitoring focused not only on primary metrics but also on indicators of distribution shift, constraint adherence, and stability of Customer 360 state evolution. Only after these assessments remain within configured bounds over sufficient time windows does the system promote the new configuration to broader coverage.

A disciplined experimentation framework also clarifies how to interpret partial or ambiguous results. Because Customer 360-centric recommenders operate in non-stationary environments, experiment outcomes may vary across cohorts or periods [41]. Implementation patterns therefore include stratified analyses conditioned on key Customer 360 features, such as tenure, activity level, or product affinity vectors, even if these features are not used for assignment. Neutral interpretation avoids overstating generality: evidence of improvement in specific strata can motivate targeted deployment for those strata while preserving incumbent policies elsewhere. The same infrastructure that provides stratified insights can later support adaptive policies that vary objective weights or constraints as functions of x_u, once appropriate guardrails are defined.

Offline replay and simulation techniques provide complementary tools for examining proposed changes before live experiments. In a replay scenario, logged requests and candidate sets from past traffic are fed to alternative ranking or policy functions that use the corresponding Customer 360 states. Because actual user responses were generated under different exposures, such replays do not furnish unbiased effect estimates, but they reveal structural properties, such as how often sensitive constraints would be invoked, how exposure distributions across items or categories would shift, and how scores correlate with observed outcomes [42]. When logging includes propensities, off-policy estimators can be applied to approximate the performance of counterfactual policies, subject to the usual assumptions. Implementation patterns encourage storing sufficient context in logs to support these analyses without repeated instrumentation changes.

Model lifecycle management patterns emerge naturally from the combination of shared embeddings, staged experiments, and replay capabilities. Each production model version is associated with metadata capturing training intervals, feature configurations drawn from Customer 360 views, objective weights, and constraint settings. Deprecation of older models is handled cautiously, especially when long-term outcome analyses rely on consistent scoring functions [43]. For some use cases, it is beneficial to maintain a small number of stable baseline models that serve as reference points in experiments, even as more adaptive models evolve. These baselines provide anchors for interpreting changes in recommendation behavior relative to known, well-characterized systems. Because all models depend on the same underlying Customer 360 abstractions, maintaining such baselines imposes limited incremental complexity.

Cross-functional alignment is an important practical dimension of implementation. Customer 360-centric recommender systems interact with product management, marketing, legal, and analytics functions that have distinct perspectives on objectives and constraints [44]. Implementation patterns therefore include configuration layers where business stakeholders can adjust interpretable parameters, such as threshold values for exposure frequencies, weights assigned to specific outcome proxies, or eligibility rules for certain content types. These parameters are translated into policy-level functions that operate on Customer 360 states and model scores without directly modifying model code. Logging of configuration states alongside events ensures that later analyses can attribute observed changes in behavior or outcomes to specific configuration choices, reducing reliance on anecdotal explanations.

Operational incident handling forms another dimension where structured implementation patterns are beneficial [45]. When anomalies arise, such as sudden drops in engagement, unexpected shifts in catalog coverage, or deviations in constraint-related metrics, runbooks guide investigation by querying Customer 360 feature distributions, model score distributions, experiment assignments, and configuration histories. These queries leverage the same unified identifiers and lineage records used for modeling, enabling systematic tracing from symptoms back to plausible sources. If necessary, rollback mechanisms revert to prior model versions or configurations while retaining all logs for post hoc analysis. This process benefits from the modularity of Customer 360-centric designs: because identity resolution, feature computation, and policy logic are separated, issues can often be isolated to one layer without indiscriminate changes elsewhere.

Organizations operating across multiple regions, brands, or product lines often confront the question of whether to maintain a single global Customer 360 representation or multiple localized variants [46]. Implementation patterns that support modularity at the schema level allow a shared core, such as identity resolution and generic behavioral features, to coexist with regional extensions that capture local attributes or regulatory constraints. Recommender models can either share parameters across regions with region indicators included in x_u, or adopt partially shared architectures where some layers are global and others are region-specific. The decision is guided by empirical assessment of transferability and by constraints on data residency and processing. By encoding these choices declaratively in the Customer 360 schema and model configuration, organizations retain flexibility without proliferating incompatible systems.

A further consideration concerns how Customer 360-centric recommenders interoperate with other decision systems, such as pricing engines, promotional targeting, or service prioritization. These systems may consume similar or overlapping features and may target related objectives [47]. Implementation patterns that treat Customer 360 as a platform resource encourage coordination: central feature stores and embedding services provide consistent inputs, while downstream systems define independent policies that can be aligned through shared monitoring. When conflicts emerge, such as recommendation policies suggesting items that contradict pricing or promotional constraints, resolution logic is applied at a policy integration layer that has visibility into all relevant signals. This layer can enforce precedence rules or compute joint decisions that respect configured hierarchies, again using Customer 360 states as the common conditioning context.

Over time, the accumulation of experiments, configuration changes, and model updates yields a substantial corpus of operational knowledge encoded in logs and metadata. Implementation patterns can exploit this corpus to inform meta-learning or automated policy search, in which higher-level systems propose candidate configurations or architectures based on historical performance under similar conditions [48]. Such automation does not obviate human oversight; instead, it provides structured suggestions that can be evaluated through the same experimental and governance framework. The Customer 360-centric perspective remains consistent: proposals are framed in terms of how they would alter mappings from unified customer states and item attributes to exposure decisions, and their evaluation relies on stable identity-resolved history.

In summary, implementation patterns and experimentation methodology for Customer 360-centric recommender systems revolve around stable data contracts, shared representation services, staged rollout and evaluation processes, modular policy configuration, and integrated observability. These patterns allow organizations to realize the benefits of unified customer representations while maintaining control over complexity and ensuring that changes in models or objectives are traceable and reversible [49]. By structuring practical decisions around the same abstractions that underlie the formal models, the technical and operational aspects of recommendation systems remain aligned, supporting incremental evolution without requiring disruptive redesign when objectives, constraints, or technologies change.

Objective Type	**Example Metric**	**Purpose in Multi-Objective Framework**
Engagement	Click or dwell probability	Captures user interest and satisfaction
Revenue	Predicted incremental sales	Optimizes short-term commercial value
Diversity	Exposure variance or coverage	Ensures catalog balance and novelty
Compliance	Fairness, risk, margin limits	Satisfies operational or policy constraints

Table 7. Representative Objective Categories in Multi-Objective Recommendation.

7. Systems Architecture, Deployment, and Monitoring

Designing Customer 360-centric recommender systems for B2C scenarios requires alignment between mathematical models and engineering architecture. The overarching constraint is that recommendations must be computed under strict latency and availability targets, often within tens of milliseconds at peak traffic, while simultaneously leveraging customer state representations that depend on large volumes of rapidly changing data.

Model Component	Notation / Function	Role in Optimization
Reward Estimator	$r_{u,i}$	Predicts immediate utility (click/purchase)
Penalty Term	$c_{u,i}$	Represents repetition or low-margin costs
Weighted Objective	$J_\lambda = \sum_k \lambda_k \mathbb{E}[R^{(k)}]$	Balances competing goals via weights λ_k
Regularizer	$\Omega_m(\theta)$	Enforces soft constraints like exposure diversity

Table 8. Mathematical Components of Multi-Objective Optimization.

Modeling Strategy	Technique	Application Context
Supervised	Cross-entropy with auxiliary penalties	Standard CTR or CVR training
Regularized	Diversity or fairness regularizers	Balances exposure across items/groups
Sequential / RL	Policy gradient, actor-critic	Long-term reward or retention focus
Hybrid	Supervised + heuristic penalties	Practical compromise for stability

Table 9. Learning Strategies for Multi-Objective Customer 360 Recommenders.

Temporal Element	Definition	Function in State Evolution
State Variable	$h_{u,t}$	Encodes customer state at time t
Transition Function	$F(h_{u,t}, e_{u,t})$	Updates latent state with new events
Scoring Function	$q(h_{u,t}, z_i)$	Computes relevance at each timestep
Survival Function	$S(t \mid x_{u,t})$	Estimates churn risk or expected value

Table 10. Temporal and Causal Elements in Customer 360-Centric Modeling.

A conceptual architecture maintains three coupled layers. The data layer manages raw event ingestion, identity resolution, and construction of Customer 360 state representations [50]. The model layer hosts training pipelines, feature transformations, and inference services. The policy layer orchestrates ranking logic, multi-objective adjustments, exploration mechanisms, and fallbacks. While implementations vary, it is useful to maintain a clear boundary where the Customer 360 store surfaces stable, versioned feature sets or embeddings to downstream models so that training and serving environments share consistent semantics [51].

Offline training pipelines consume historical logs with explicit exposure and outcome fields to construct supervised and counterfactual training sets. Features include Customer 360 embeddings x_u, item representations z_i, and contextual variables. Models are retrained on sliding windows to accommodate drift, and training configurations are versioned to support rollback and audit [52]. To avoid divergence between offline and online behavior, feature generation code paths are shared as much as possible, with only the minimal necessary optimizations in the serving path, such as approximate nearest neighbor indices for candidate retrieval.

Evaluation Method	Estimator / Process	Strengths and Limitations
Randomized Experiment	A/B or contextual bandit	Unbiased but costly, requires stable IDs
Inverse Propensity Weighting	$\hat{V}(\pi)$ estimator	Enables offline evaluation, sensitive to low propensities
Doubly Robust	Combined outcome + propensity models	Reduces variance, needs accurate estimation
Simulation / Replay	Logged traffic re-evaluation	Reveals structure, not causal effects

Table 11. Evaluation Techniques for Temporal and Causal Analysis.

Implementation Pattern	Description	Key Benefit
Canonical Feature Views	Stable aggregates and sequences	Enable cross-model consistency
Shared Embedding Services	Centralized x_u, z_i models	Promote reuse and version control
Experimentation Framework	Customer-level assignment	Preserves temporal and identity coherence
Shadow Rollouts	Offline and dual-writing evaluation	Safe validation before deployment
Configuration Layers	Business-adjustable parameters	Supports transparency and governance

Table 12. Common Implementation Patterns in Customer 360-Centric Recommenders.

Online inference services are typically structured into candidate generation and ranking stages. Candidate generation uses lightweight models to identify a manageable subset of items, using functions that may approximate $x_u^\top z_i$ or related similarities. Ranking models then evaluate richer signals and apply multi-objective scores. Within this structure, Customer 360-centric design ensures that both stages operate on the same underlying customer representation, thereby preventing contradictions where candidate generation is based on incomplete or inconsistent views relative to rankers. Latency budgets are allocated across stages, and caching strategies are informed by the temporal stability of x_u and z_i.

Monitoring is essential for detecting drift, regressions, and unintended consequences [53]. Core metrics include engagement, conversion, revenue, and stability indicators such as variance in exposure across segments or items. In addition, diagnostic metrics track distributions of x_u and model scores over time. For example, sudden shifts in embedding norms or score histograms can indicate upstream data issues or feedback loops. Quantitative guards, such as thresholds on predicted probabilities or entropy constraints on exposure distributions, can be enforced at the policy layer, decoupled from the core prediction models.

Exploration mechanisms integrate with Customer 360 design by using per-customer states to control experimentation exposure and prevent over-saturation or starvation of certain groups [54]. Contextual bandit strategies allocate traffic among candidate policies or recommendation templates while ensuring that sufficient support exists across relevant strata. Logged propensities from these mechanisms support the counterfactual estimators previously discussed. The architecture thus links exploration, evaluation, and model updates in a closed loop, with Customer 360 states providing the common reference frame.

Robustness and governance considerations complete the design. Access controls regulate which components can read or modify Customer 360 features [55]. Audit logs document when model configurations, objective weights, or feature subsets change. When regulatory constraints require exclusion of specific attributes or inference types, the system can implement these at the feature layer so that downstream models function on compliant inputs without uncoordinated adjustments. In this setting, technical neutrality and configurability are central: the architecture does not bake in normative judgments but makes explicit the levers through which organizations can align recommendations with applicable policies.

8. Conclusion

This paper has outlined a technical perspective on designing Customer 360-centric recommender systems intended to optimize B2C digital sales and engagement outcomes within a coherent machine learning framework. The central premise is that unified, temporally aware customer representations should serve as primary inputs to recommendation policies, rather than as auxiliary constructs confined to reporting or segmentation [56]. By grounding modeling efforts in a well-defined Customer 360 data layer, it becomes possible to build recommendation pipelines that systematically integrate heterogeneous signals, support multi-objective optimization, and accommodate operational and governance constraints.

The discussion has emphasized several elements that interact in practice. First, robust Customer 360 data foundations depend on reliable identity resolution, explicit handling of domain heterogeneity, and alignment between offline and online feature semantics. Second, representation learning and feature fusion methods provide compact yet expressive ways to encode trajectories and attributes into embeddings suitable for high-throughput recommendation, while preserving adaptability under distribution shift. Third, multi-objective modeling formulates engagement, revenue, and additional constraints within a shared optimization problem, making trade-offs explicit and tunable rather than implicit [57]. Fourth, temporal dynamics, causal structure, and counterfactual evaluation are treated not as peripheral considerations but as intrinsic to responsible policy updates in settings where historical data reflect evolving exposure mechanisms. Finally, systems architecture and monitoring close the loop between modeling decisions and production realities, incorporating latency, reliability, auditability, and configurable control.

The resulting view does not assume a single optimal architecture or model. Instead, it characterizes a set of interdependent design choices through which organizations can construct Customer 360-centric recommender systems that are technically consistent with their data, objectives, and constraints. Subsequent empirical work in specific domains can instantiate these abstractions, quantify the behavior of concrete implementations, and refine modeling components where necessary without altering the underlying structural considerations. [58]

References

[1] T. Cao and H. Xie, "A literature review of supply chain development: Evidence from agricultural industry," *BCP Business & Management*, vol. 20, pp. 453–462, 6 2022.

[2] F. Ulbrich, T. Christensen, and L. Stankus, "Gender-specific on-line shopping preferences," *Electronic Commerce Research*, vol. 11, no. 2, pp. 181–199, 11 2010.

[3] F. Hackl, M. Hölzl-Leitner, R. Winter-Ebmer, and C. Zulehner, "Successful retailer strategies in price comparison platforms," *Managerial and Decision Economics*, vol. 42, no. 5, pp. 1284–1305, 3 2021.

[4] J. Angelopoulos and D. Mourtzis, "An intelligent product service system for adaptive maintenance of engineered-to-order manufacturing equipment assisted by augmented reality," *Applied Sciences*, vol. 12, no. 11, pp. 5349–5349, 5 2022.

[5] M. A. ul Haq, F. Akram, and H. A. M. Malik, "The economics of renewable energy expansion for rural households," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 269–277, 7 2022.

[6] E. Harinurdin, "Optimization implementation of foreign customer information submission system (sipina)," *KnE Social Sciences*, vol. 3, no. 11, pp. 404–417, 8 2018.

[7] L. Pasape, "The influence of information and communication technology on the business performance of the incubated small business enterprises in tanzania," *Journal of Economics, Management and Trade*, pp. 30–46, 5 2022.

[8] F. Destari, K. Indraningrat, and M. N. N. Putri, "Impact of shopping emotion towards impulse buying in e-commerce platform," *Jurnal Manajemen dan Pemasaran Jasa*, vol. 13, no. 1, pp. 47–64, 5 2020.

[9] S. H. Kukkuhalli, "Enabling customer 360 view and customer touchpoint tracking across digital and non-digital channels," *Journal of Marketing & Supply Chain Management*, vol. 1, no. 3, 2022.

[10] R. Kumar., "The future of online shopping in india. a study of punjab and haryana states of india." *International Journal of Advanced Research*, vol. 4, no. 5, pp. 1528–1544, 5 2016.

[11] H. C. Gamper, "How can internet comparison sites work optimally for consumers," *Journal of Consumer Policy*, vol. 35, no. 3, pp. 333–353, 5 2012.

[12] I. RHAROUBI and K. TALMENSSOUR, "Understanding the role of supply chain digitalization in the quality of buyer-supplier relationship: case studies of moroccan companies," *Zenodo (CERN European Organization for Nuclear Research)*, 7 2022.

[13] P. Jain and G. Bansal, "Study on effects of shopping orientation on the consumers buying online in punjab," *Bhatter College Journal of Multidisciplinary Studies*, vol. 7, no. 1, 6 2017.

[14] M. Binsawad, "Social media efficiency towards restaurant business: a comparison between social media profiles (case study in saudi arabia)," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 31 389–31 399, 8 2020.

[15] E. Gurgu, I.-A. Gurgu, and R. Tonis, "Neuromarketing for a better understanding of consumer needs and emotions," *Independent Journal of Management & Production*, vol. 11, no. 1, pp. 208–235, 2 2020.

[16] L. M. Damarwulan, "E-wom bomb effect on social media influence to brand: Cases in halal products," *Journal of Management and Business*, vol. 14, no. 1, 3 2015.

[17] P. Valchuk, A. Bespalykh, and S. Kivi, "Dynamicallydevelopingmarkets:outlookofcurrenttrendsandimplicationsforbusinessprocesses," *EurasianUnionScientists*, vol. 1, no. 5(74), pp. 6–10, 6 2020.

[18] V. T. Dang, J. Wang, and T. T. Vu, "An integrated model of the younger generation's online shopping behavior based on empirical evidence gathered from an emerging economy." *PloS one*, vol. 15, no. 5, pp. e0 232 213–, 5 2020.

[19] N. Adamashvili and M. Fiore, "Investigating the role of business marketing techniques in sales process," *European Journal of Management Issues*, vol. 25, no. 3, pp. 135–143, 12 2017.

[20] R. Gusnita, "Analysis of customer satisfaction in the online shopping system for fashion product," *Inovbiz: Jurnal Inovasi Bisnis Seri Manajemen, Investasi dan Kewirausahaan*, vol. 1, no. 2, pp. 76–76, 12 2021.

[21] R. M. Kumar, "A study on customer engagement and loyalty towards the digital marketing," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 9, pp. 1079–1087, 9 2019.

[22] L. Qinzhe, T. Bingbing, L. Dongliang, L. Yan, F. Yuhong, L. Chen, and K. Zhao, "An integrated energy service transaction model based on energy blockchain," *International Journal of Heat and Technology*, vol. 38, no. 2, pp. 293–300, 6 2020.

[23] M. T. Bulan and S. Sukesi, "Analysis of the effect of service quality, price and perceptions of risk online shopping against purchase interest in e-commerce customers pt. matahari department store tbk kupang branch," *Ekspektra : Jurnal Bisnis dan Manajemen*, vol. 4, no. 1, pp. 45–64, 6 2020.

[24] S. Miah and T. K. Das, "Behavioral patterns of stakeholders in online business in bangladesh: A qualitative exploration," *International Journal of Business Anthropology*, vol. 11, no. 2, 12 2021.

[25] J. Kropivšek, P. Grošelj, L. Oblak, and M. Jošt, "A comprehensive evaluation model for wood companies websites based on the ahp/r-topsis method," *Forests*, vol. 12, no. 6, pp. 706–, 5 2021.

[26] H.-K. Kong, T.-S. Kim, and J. Kim, "An analysis on effects of information security investments: a bsc perspective," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 941–953, 4 2010.

[27] B. Maas, "Literature review of mobility as a service," *Sustainability*, vol. 14, no. 14, pp. 8962–8962, 7 2022.

[28] H. Arya, "E-banking: The emerging trend," *International Journal of Trend in Scientific Research and Development*, vol. Volume-3, no. Issue-4, pp. 449–455, 6 2019.

[29] V. Sales-Vivó, I. Gil-Saura, and M. G. Gallarza, "Value co-creation and satisfaction in b2b context: A triadic study in the furniture industry," *Sustainability*, vol. 13, no. 1, pp. 152–, 12 2020.

[30] A. S. Subroto, "Study on indonesia consumer trust on online transaction," *International Journal of Business Studies*, vol. 1, no. 2, pp. 41–50, 9 2018.

[31] J. Zhao and H. Zhao, "Design of prototype system for multi-agent supply chain information sharing benefit distribution management," *Information Systems and e-Business Management*, vol. 18, no. 4, pp. 581–602, 12 2018.

[32] L. Ashworth and C. Free, "Marketing dataveillance and digital privacy: Using theories of justice to understand consumers' online privacy concerns," *Journal of Business Ethics*, vol. 67, no. 2, pp. 107–123, 8 2006.

[33] S. A. Ehikioya and J. Zeng, "Mining web content usage patterns of electronic commerce transactions for enhanced customer services," *Engineering Reports*, vol. 3, no. 11, 5 2021.

[34] G. F. Marias, J. Barros, M. Fiedler, A. Fischer, H. Hauff, R. Herkenhoener, A. Grillo, A. Lentini, L. Lima, C. Lorentzen, W. Mazurczyk, H. de Meer, P. F. Oliveira, G. C. Polyzos, E. Pujol, K. Szczypiorski, J. P. Vilela, and T. T. V. Vinhoza, "Security and privacy issues for the network of the future," *Security and Communication Networks*, vol. 5, no. 9, pp. 987–1005, 10 2011.

[35] L. Gill-Simmen, D. J. MacInnis, A. B. Eisingerich, and C. W. Park, "Brand-self connections and brand prominence as drivers of employee brand attachment," *AMS Review*, vol. 8, no. 3, pp. 128–146, 3 2018.

[36] T. Mehmood, "Does information technology competencies and fleet management practices lead to effective service delivery? empirical evidence from e- commerce industry," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 1, no. 2, pp. 14–41, 12 2021.

[37] S. H. Kukkuhalli, "Improving digital sales through reducing friction points in the customer digital journey using data engineering and machine learning," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, vol. 10, no. 3, 2022.

[38] M. Marolt, A. Pucihar, and H.-D. Zimmermann, "Social crm adoption and its impact on performance outcomes : a literature review," *Organizacija*, vol. 48, no. 4, pp. 260–271, 12 2015.

[39] S. U. Deshmukh, "Impact of e-business on business association," *International Journal of Engineering and Management Research*, vol. 9, no. 6, pp. 7–12, 12 2019.

[40] M. Gerner, "Assessing and managing sustainability in international perspective: corporate sustainability across cultures – towards a strategic framework implementation approach," *International Journal of Corporate Social Responsibility*, vol. 4, no. 1, pp. 1–34, 6 2019.

[41] J. Yu, J. Wang, and T. Moon, "Influence of digital transformation capability on operational performance," *Sustainability*, vol. 14, no. 13, pp. 7909–7909, 6 2022.

[42] I. Ahmedov, "The impact of digital economy on international trade," *European Journal of Business and Management Research*, vol. 5, no. 4, 7 2020.

[43] J.-Y. Lai, K. R. Ulhas, and J.-D. Lin, "Assessing and managing e-commerce service convenience," *Information Systems Frontiers*, vol. 16, no. 2, pp. 273–289, 2 2012.

[44] Štefan Slávik, I. M. Hudáková, K. Procházková, and B. Zagoršek, "Strategic background of the start-up—qualitative analysis," *Administrative Sciences*, vol. 12, no. 1, pp. 17–17, 1 2022.

[45] N. Šimková and Z. Smutny, "Business e-negotiation: A method using a genetic algorithm for online dispute resolution in b2b relationships," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 16, no. 5, pp. 1186–1216, 3 2021.

[46] J. Llopis, M. R. González, and J. L. Gascó, "Transforming the firm for the digital era: an organizational effort towards an e-culture," *Human Systems Management*, vol. 23, no. 4, pp. 213–225, 12 2004.

[47] C.-L. Chen, Y.-C. Lin, W.-H. Chen, C.-F. Chao, and H. Pandia, "Role of government to enhance digital transformation in small service business," *Sustainability*, vol. 13, no. 3, pp. 1028–, 1 2021.

[48] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," *Electronics*, vol. 11, no. 1, pp. 141–141, 1 2022.

[49] X. Zhang, X. Ming, Z. Liu, D. Yin, Z. Chen, and Y. Chang, "A reference framework and overall planning of industrial artificial intelligence (i-ai) for new application scenarios," *The International Journal of Advanced Manufacturing Technology*, vol. 101, no. 9, pp. 2367–2389, 11 2018.

[50] G. Ilieva, T. Yankova, S. Klisarova, and Y. Dzhabarova, "Customer satisfaction in e-commerce during the covid-19 pandemic," *Systems*, vol. 10, no. 6, pp. 213–213, 11 2022.

[51] S. H. Kukkuhalli, “Increasing digital sales revenue through 1:1 hyper-personalization with the use of machine learning for b2c enterprises,” *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 1, no. 1, 2023.

[52] M. Park, J. U. Min, and S.-Y. Lee, “Implications of growing electronic commerce for freight transportation: A case study of the united states,” *Journal of International Logistics and Trade*, vol. 4, no. 2, pp. 37–52, 12 2006.

[53] Y.-C. Chou and B. B. M. Shao, “Adoption and performance of mobile sales channel for e-retailers: Fit with m-retail characteristics and dependency on e-retailing,” *Information Systems Frontiers*, vol. 23, no. 3, pp. 681–694, 2 2020.

[54] P. Jia and C. Stan, “Artificial intelligence factory, data risk, and vcs’ mediation: The case of bytedance, an ai-powered startup,” *Journal of Risk and Financial Management*, vol. 14, no. 5, pp. 203–, 5 2021.

[55] R. Yuliantoro, T. Y. R. Syah, S. Pusaka, and D. Hs, “Implementation of marketing mix strategy for start-up business: fruit combining,” *Russian Journal of Agricultural and Socio-Economic Sciences*, vol. 87, no. 3, pp. 220–230, 3 2019.

[56] I. Biclesanu, S. Anagnoste, O. Branga, and M. Savastano, “Digital entrepreneurship: Public perception of barriers, drivers, and future,” *Administrative Sciences*, vol. 11, no. 4, pp. 125–, 11 2021.

[57] C. Ivan and R. Popa, “A cloud based mobile dispatching system with built-in social crm component: Design and implementation,” *Computers*, vol. 4, no. 3, pp. 176–214, 7 2015.

[58] E. Lorenzini, “Innovation and e-commerce in clusters of small firms: The case of a regional e-marketplace:,” *Local Economy: The Journal of the Local Economy Policy Unit*, vol. 29, no. 8, pp. 771–794, 10 2014.